LA-UR- 02-2478

*Approved for public release;
distribution is unlimited.*

Title:

SENSITIVITY ANALYSIS OF A NONLINEAR NEWTON-KRYLOV SOLVER FOR HEAT TRANSFER WITH PHASE CHANGE

Author(s):

Rudolph J. Henninger (CCS-2), Dana A. Knoll (T-3), Douglas B. Kothe (CCS-2), and Bryan Lally (CCS-2)

Submitted to:

WCCM V - Fifth World Congress on Computational Mechanics
Vienna, Austria
July 7th - 12th, 2002

WCCM V
Fifth World Congress on
Computational Mechanics
July 7-12, 2002, Vienna, Austria
Eds.: H.A. Mang, F.G. Rammerstorfer,
J. Eberhardsteiner

Sensitivity Analysis of a Nonlinear Newton-Krylov Solver for Heat Transfer with Phase Change

Rudolph J. Henninger*, Dana A. Knoll (T-3), Douglas B. Kothe and Bryan Lally

Los Alamos National Laboratory
Group CCS-2, Mail Stop D413, Los Alamos, NM 87544, USA
e-mail: rjh@lanl.gov

Key words: sensitivity analysis, automatic differentiation, CFD, casting

Abstract

Development of a complex metal-casting computer model requires information about how varying the problem parameters affects the results (metal flow and solidification). For example, we would like to know how the last point to solidify or the cooling rate at a given location changes when the physical properties of the metal, boundary conditions, or mold geometry are changed. As a preliminary step towards a complete sensitivity analysis of a three-dimensional casting simulation, we examine a one-dimensional version of a metal-alloy phase-change conductive-heat-transfer model by means of Automatic Differentiation (AD). This non-linear "Jacobian-free" method is a combination of an outer Newton-based iteration and an inner conjugate gradient-like (Krylov) iteration. The implicit solution algorithm has enthalpy as the dependent variable from which temperatures are determined. We examine the sensitivities of the difference between an exact analytical solution for the final temperature and that produced by this algorithm to the problem parameters. In all there are 17 parameters (12 physical constants such as liquid density, heat capacity, and thermal conductivity, 2 initial and boundary condition parameters, the final solution time, and 2 algorithm tolerances). We apply AD in the forward and reverse mode and verify the sensitivities by means of finite differences. In general, the finite-difference method requires at least N+1 computer runs to determine sensitivities for N problem parameters. By forward and reverse, we mean the direction through the solution and in time and space in which the derivative values are obtained. The forward mode is typically more efficient for determining the sensitivity of many responses to one or a few parameters, while the reverse mode is better suited for sensitivities of one or a few responses with respect to many parameters. The sensitivities produced by all the methods agreed to at least three significant figures. The forward and reverse AD code run times were similar and were approximately 34% faster than those of the finite-difference sensitivities. Real problems in three dimensions will certainly have many more parameters describing mold geometry and pouring conditions. If the trend seen here holds true reverse mode AD is favored since the computational time increases only slightly for additional parameters.

1 Introduction

The Accelerated-Strategic-Computing-Initiative (ASCI) Telluride [1] project is tasked with developing and applying a new simulation tool designed to model and optimize the gravity-pour casting processes which are currently ongoing at Los Alamos National Laboratory foundries in support of the U.S Department of Energy (DOE) alloy manufacturing needs. This tool, known as Telluride, is a new software package that is being written in Fortran 90 for all high-performance ASCI computing platforms. The Telluride simulation tool must accurately model the entire alloy casting process in one integrated simulation. This process begins with molten-alloy free-surface flow characteristic of the pouring and filling of a mold cavity. Cooling and solidification of the alloy follows shortly thereafter, followed by a more gradual cooling of the solid alloy to room temperature, which completes the casting process. Further homogenization heat treatment and machining prepare the alloy part for integration into the engineering system for which it is designed. Telluride must embody realistic physical models for alloy solidification, micro/macro segregation phenomena, free surface incompressible flow, heat transfer, complex 3-D geometric effects, microstructural nucleation and evolution, solid state transition effects, and residual stress buildup and response. Initially Telluride must help foundry workers understand and optimize their current casting processes, but ultimately Telluride must facilitate a priori design of new casting processes tuned for alloy components possessing specific microstructural properties. Realistic, macroscopic physical models must be developed for a wide variety of phenomena: incompressible turbulent free surface flow, conductive/convective/radiative heat transfer, alloy microstructure evolution and solidification, macro segregation of alloy species, solid/solid phase change, residual stress and distortion, and heat treatment, to name a few. The coupling and interplay of all these phenomena is arguably the most difficult task, as their simultaneous occurrence during a typical casting process is commonplace. Physical model development efforts to date have focused on those phenomena most important prior to solid/solid phase transition and subsequent stress buildup, e.g., the first four phenomena listed above. Efforts are currently focused on those phenomena important after first solidification, i.e., from first solid, through solid phase transitions, cooled down to room temperature.

Accurate, convergent, and robust discrete numerical solutions to the set of coupled, nonlinear PDEs arising from models describing these complex phenomena are needed on complex, 3-D computational domains partitioned with generally unstructured computational cells. A temporally and spatially second order finite volume discretization amenable to arbitrary element types is in place and is serving as an excellent testbed for high fidelity algorithms. A fully coupled nonlinear solution algorithm offered by a Newton-Krylov approach has replaced the original fractional, time-step-split flow algorithm. Additional effort is needed to assure the fidelity and robustness of the more recent implementations.

In anticipation of using this code for casting model optimization, fitting of experimental data, and uncertainty quantification we are developing the means to provide information about how varying the code input parameters affects the results (metal flow and solidification). As a preliminary step towards a complete sensitivity analysis, we examine a one-dimensional prototype version [2] of the Telluride phase-change model by means of Automatic Differentiation (AD) [3]. In the sections that follow we will provide a brief description of the phase-change model, the ADIFOR code and how it is applied, and the resulting sensitivities. We will examine the accuracy and computational effort in producing the sensitivities. We then demonstrate the use of the sensitivities as a gradient in a data fitting (optimization) exercise. Finally, we summarize our findings and outline our plans for future work.

2 Phase Change Model

Eutectic alloys freeze as a mechanical mixture of solid phases. Solidification begins when the liquidus temperature T_l is reached. When the alloy cools to just above the eutectic temperature T_{eut} the freezing begins via a eutectic reaction at a constant temperature. Therefore, the temperature condition for phase change (the so-called mushy zone) is: $T_{eut} < T < T_l$. In solid-solution alloy systems in which addition of solute lowers the freezing temperature T_f the slope of the liquidus line m_l is negative or the equilibrium phase partition coefficient $k = C_s / C_l$ (the ratio of solute concentration in the solid phase to solute concentration in the liquid phase) is less than unity. Assuming a constant liquidus slope, we have the following temperature-concentration relationship: $T = T_f + m_l C_l$. The solute is soluble at all concentrations. Therefore, the temperature condition for phase change is: $T < T_l$. In solid-solution alloy systems in which addition of solute increases the freezing temperature m_l is positive or k is greater than unity. The solute is assumed to be soluble at all concentrations. The temperature condition for phase change is: $T_f < T < T_l$. Telluride utilizes a new nonlinear solution method for all heat transfer problems. This approach yields a fully implicit solution of the enthalpy form of the energy equation. By doing so, Telluride can both accurately evolve the latent heat in an isothermal solidification process and also couple the temperature and concentration fields in the nonisothermal solidification of multicomponent alloys. Implicit solution algorithms are often preferred to avoid undesirable time step restrictions. Telluride presents a fully implicit method for the enthalpy form of the energy equation. Enthalpy is the dependent variable in our nonlinear iterations, so there is no local change of dependent variables and the solidifying front evolves in an accurate, self-consistent manner. Energy is rigorously conserved. In pure material situations, this method captures the solidification front accurately over a wide range of time step sizes; tracking is not required. In binary eutectic alloy problems, capturing the front with an implicit enthalpy form of the energy equation allows for integration to be performed over large time steps without sacrificing accuracy.

To summarize the solution technique (details are provided in [2]): we solve the discrete form of the one-dimensional enthalpy (H_i^{n+1}) equation for cell i at time step $n+1$ by iteratively minimizing

$$F_i = \frac{H_i^{n+1} - H_i^n}{\Delta t} - \frac{\kappa}{\rho \Delta x^2}\left[\tau\left(H_{i+1}^{n+1}\right) - 2\tau\left(H_i^{n+1}\right) + \tau\left(H_{i-1}^{n+1}\right)\right] \tag{1}$$

where κ is the thermal conductivity, ρ is the density, Δt is the time step, Δx is the spatial mesh size, and

$$\tau(H) = \begin{cases} H / C_p, & H < C_p T_{eut}; \\ T_{eut}, & C_p T_{eut} \le H \le C_p T_{eut} + \left(1 - \varepsilon_{s,eut}\right) L; \\ \left(H - \left(1 - \varepsilon_s\right) L\right) / C_p, & C_p T_{eut} + \left(1 - \varepsilon_{s,eut}\right) L \le H \le C_p T_l + L; \\ \left(H - L\right) / C_p, & H > C_p T_l + L \end{cases} \tag{2}$$

Where C_p is the specific heat, ε_s is the solid volume fraction, and L is the latent heat. If the Scheil temperature-solid volume fraction relationship is used, the additional nonlinear system

$$C_p T + \left(1 - \varepsilon_s\right) L - H = 0 \qquad (3)$$

$$\varepsilon_s + \left(\frac{T_f - T_l}{T_f - T}\right)^{-(1-k)} - 1 = 0 \qquad (4)$$

is solved for the cells within the mushy zone by a standard Newton iteration. For each time step, an inexact Newton-GMRES (Generalized Minimal RESidual) iteration is used to solve the nonlinear system arising from the implicit discretization of an enthalpy based energy equation (Eq. 1). This inexact Newton-GMRES iteration achieves Newton-like nonlinear convergence solely through a special matrix-vector multiply routine (**matvec**) in GMRES. The true Jacobian of the system is never formed and never inverted. We express the implicit algorithm in template form:

```
Time Step Loop ( n is time step index )
        $\mathbf{H}^{n+1} = \mathbf{H}^{n}$ (initial guess)
        Form preconditioning matrix, M (Eq. 5)
                while (not converged) do ( Newton Loop)
                ( k is nonlinear iteration index)
                If k = 1 form $F(\mathbf{H}^{k})$
                        while (not converged) do (GMRES Loop)
                        (m is linear iteration index)
                        solving $(\mathbf{J}^{k}M^{-1})(M\,\delta\mathbf{H}^{k}) = F(\mathbf{H}^{k})$ for $\delta\mathbf{H}^{k}$
                        Initial Krylov vector, $\mathbf{v}_{0} = F(\mathbf{H}^{1})$
                        (build $\mathbf{v}_{m} = \mathbf{J}^{k}M^{-1}\mathbf{v}_{m-1}$ , m th Krylov vector )
                                preconditioning operation, iteratively solve $M\mathbf{y} = \mathbf{v}_{m-1} \Rightarrow \mathbf{y}$
                                (fixed number of Symmetric Gauss-Seidel iterations)
                                matvec operation => $\mathbf{J}^{k}\mathbf{y} \approx [F(\mathbf{H}^{k} + \varepsilon\mathbf{y}) - F(\mathbf{H}^{k})]/\varepsilon$
                        Complete m^th GMRES iteration, compute $\boldsymbol{\beta}_{j}$ s (Eq. 6)
                        Linear Convergence ?
                        end (GMRES Loop)
                Evaluate Newton damping scalar, $d$ [so max $\delta\mathbf{H} < 10\%$]
                $\mathbf{H}^{k+1} = \mathbf{H}^{k} + d\,\delta\mathbf{H}^{k}$
                Evaluate $F(\mathbf{H}^{k+1})$, Nonlinear Convergence ?
                end (Newton Loop)
        $\mathbf{H}^{n+1} = \mathbf{H}^{k+1}$
        time = time + Δt
end (Time Step Loop)
```

$$\left.\begin{aligned} M_{i,i-1} &= -\frac{\kappa}{\rho C_p \Delta x^2} \\ M_{i,i} &= \frac{1}{\Delta t} + 2\frac{\kappa}{\rho C_p \Delta x^2} \\ M_{i,i+1} &= -\frac{\kappa}{\rho C_p \Delta x^2} \end{aligned}\right\} \tag{5}$$

$$\delta \mathbf{H}_m = \delta \mathbf{H}_0 + \sum_{j=0}^{m-1} \beta(\mathbf{J})^j \mathbf{v}_0 \tag{6}$$

For a detailed GMRES template consult reference [4].

3 Automatic Differentiation

The AD program ADIFOR [3] was applied to the FORTRAN77 code that implemented the solver technique described in Section 2 in order to provide sensitivities. ADIFOR was developed jointly in a collaboration of Argonne National Laboratory and Rice University, and its acronym comes from Automatic DIfferentiation of FORtran. All AD codes have two stages in getting from the original code to an executable code with derivative coding included. The first step is to submit the original code to a precompiler. This precompiler analyzes the code and augments it with statements for the computation of the derivatives of interest based on the chain rule of differential calculus. The output of this step is enhanced code, with some calls to external ADIFOR subroutines (for storage and memory manipulation, usually). ADIFOR gives no source code for the pre-compiler, limiting the user to given executables for the Sun, IBM, RS6000, SGI, LINUX, and Alpha workstations. For executing the augmented code, any machine upon which the libraries can be compiled properly is available. Once the original code has been augmented, this code is compiled and loaded with ADIFOR libraries, and executable derivative (sensitivity) code is produced. ADIFOR requires the presence of external files in order to obtain information about the parameters and responses of interest, as well as naming all of the files that comprise the original code. It also requires that there be a main part of the program in which no computation takes place, and from which the parameters and responses are passed to the subroutines. The main or driver part of the program is not augmented, and significant modifications are necessary for it to compile with the augmented code. These include initialization of the derivative values for the parameters, as well as dimensioning of several new arrays that the augmented code has added and modifying the subroutine call statement, as well as inserting code to print out the gradients of the responses with respect to the parameters. ADIFOR currently works only on FORTRAN77 code. ADIFOR has a version for C, known as ADIC.

AD methods can be implemented in either the forward or adjoint mode. By forward and adjoint, we mean the direction through the code and in time in which the derivative values are obtained. A forward mode of differentiation involves determining the necessary derivatives by following the code logic in the forward direction (and forward in time), while for the adjoint mode, the derivatives are determined by following the code logic in the reverse direction (and backward in time). Which of these is more useful and efficient depends on the relative numbers of input parameters of interest and responses of interest. The forward mode is more efficient for determining the sensitivity of many responses to one or a few input parameters, while the adjoint mode is better suited for sensitivities of one or a few responses with respect to many input parameters.

For an optimization (data fitting) process the response of interest is a so-called cost function that, for example, computes the sum of the squares of the difference between the calculated and a desired data set. There may be many parameters of interest (these could be the material model parameters and the initial and boundary conditions). For this problem one would choose the adjoint mode, which is most efficient for one response and many input parameters. ADIFOR can be implemented in either the forward or adjoint mode. If any of the processes are non-linear as they are for the solidification solver, information from the forward calculation is needed in the adjoint calculation. The need for this information can be satisfied one of two ways: by recomputation of forward results in the adjoint calculation, or by independent storage.

To check the ADIFOR-produced sensitivities finite difference (FD) values were also produced. To obtain FD sensitivities one must carefully choose a perturbation fraction. Examination of FD sensitivities shows that large fractions lose accuracy and small fractions display subtractive cancellation errors. Here we used $1.0*10^{-7}$. A method not tried here called a complex-step derivative approximation [5] does not display subtractive cancellation errors. It produces derivatives by using an imaginary perturbation and complex arithmetic.

4 Results

We examine the sensitivities of the difference between an exact analytical solution for the final temperature and that produced by this algorithm to the problem parameters. In all there are 17 parameters (12 physical constants such as liquid density, heat capacity, and thermal conductivity, 2 initial conditions, the final solution time, and 2 algorithm tolerances). We apply AD in the forward and adjoint mode and verify the sensitivities by means of finite differences (FD). The finite-difference method requires at least N+1 computer runs to determine sensitivities for N problem parameters. We consider a 1-D binary eutectic problem, which has been adapted from reference [6] in the region 0 ≤ x ≤ 0.4 (divided into 50 equally spaced finite volumes). This problem is run with the physical constants, initial and boundary conditions, and algorithm tolerances listed in Table 1. At time zero the alloy is at 905.2 K, the left boundary is set to and held at a temperature of 621.2 K , and solidification proceeds.

Table 1 also lists the AD and FD sensitivities of the response:

$$R = \sqrt{\sum_{i=1}^{N} \left(\frac{T_i - T_i^{base}}{T_{eut}} \right)^2} \qquad (7)$$

to the problem parameters. T_i is the calculated final temperature, T_i^{base} is the 'exact' solution provided in [6], and T_{eut} is the eutectic temperature. The AD forward and adjoint values agreed to the number of digits listed in the table. It can also be seen that the AD and FD sensitivities agreed to at least 4 digits giving us confidence in the methods.

If we normalize the sensitivities we find that the most important parameter is the left boundary condition followed, respectively by the liquidus temperature, the thermal conductivity, the liquid density, and the latent heat. An important consideration is the run time for determining the sensitivities.

Table 1: Binary-eutectic-solidification problem parameters and their sensitivities

Parameter	AD Sensitivity	FD Sensitivity
C_p (1000 J/kg-K) specific heat	1.1645954E-04	1.1645954E-04
L (4.0e05 J/kg) heat of fusion	8.2645236E-07	8.2645265E-07
k (0.15) partition coefficient	3.1184820E-02	3.1184843E-02
m_l (3.4) slope of liquidus line	0.0000000E+00	0.0000000E+00
ρ_l (2400 kg/m^3)	1.8626687E-04	1.8626693E-04
ρ_s (3120 kg/m^3)	0.0000000E+00	0.0000000E+00
T_f (921.2 K) pure solvent melting temp.	-8.5837549E-04	-8.5835420E-04
T_{eut} (821.2 K) eutectic temperature	-8.5760228E-04	-8.5760060E-04
T_l (904.2 K) liquidus temperature	1.6651385E-03	1.6653906E-03
c_{l0} (5%) initial solute concentration	0.0000000E+00	0.0000000E+00
θ (1.) residual energy weight	8.9702155E-03	8.9702153E-03
Δt_∞ (10) timestep	-4.4704048E-02	-4.4704029E-02
Initial temperature (905.2 K)	1.3331557E-03	1.3333211E-03
Left boundary temperature (621.2 K)	3.1830893E-03	3.1831320E-03
κ (100 W/m-K) thermal conductivity	-4.4704048E-03	-4.4704029E-03
γ (0.01) inexact Newton tolerance	0.0000000E+00	0.0000000E+00
Norm of nonlinear residual (1.0E-5)	0.0000000E+00	0.0000000E+00

Table 2 compares the run times for various methods. The hybrid method is described below.

Table 2: Binary-eutectic-solidification problem runtimes on a PC running under LINUX

Calculation	CPU time (s)	Hybrid CPU time (s)
Single forward	2.1	-
Finite difference (FD)	36.9	-
ADIFOR forward	24.0	11.8
ADIFOR adjoint	24.7	12

We see that AD produces the sensitivities approximately 34 % faster. If we examine the algorithm results carefully we find that there are typically 4 to 6 Newton iterations for each time step. Further, we note that the first 2 iterations do not contribute much to the final value of the enthalpy. To reduce the computational effort we therefore modified the code (the so-called hybrid code) to skip the derivative calculations until the third Newton iteration. As can be seen in Table 2 this cut the run time roughly in half. The AD sensitivities produced by the hybrid code agreed to 5 digits which, depending upon their application will probably be sufficient.

It is instructive to examine the major dependent variable (enthalpy) and its adjoint as a function of space and time (Fig. 1). We see the cooling and solidification (the light blue flat area) wave proceeding into the alloy as time proceeds. The adjoint enthalpy is the importance of enthalpy to fitting the exact final temperature distribution. Thus, we see that the enthalpy near the phase transition region near the end of the calculation has the largest influence on matching the final temperature.

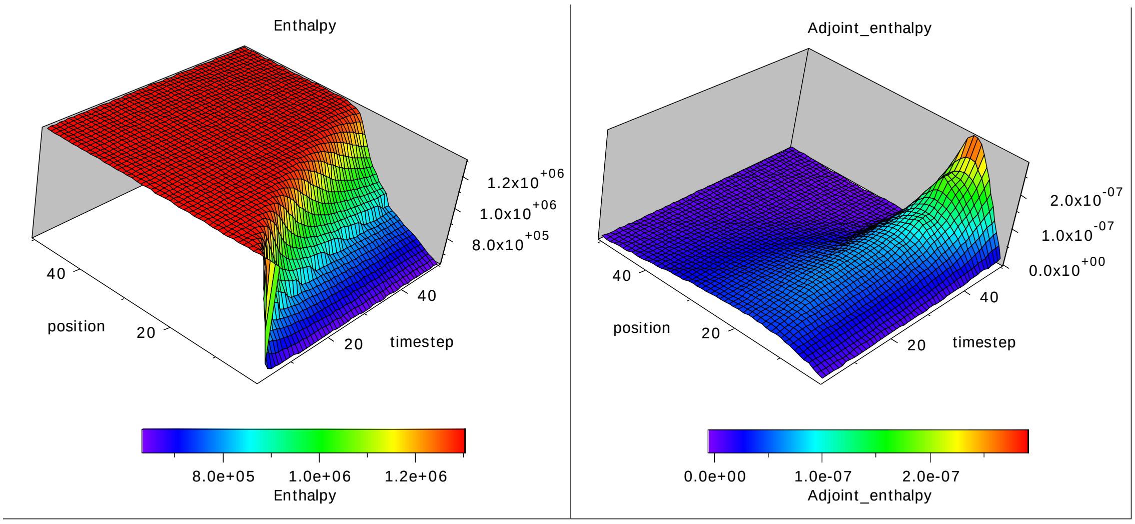

Figure 1: Enthalpy and Adjoint-Enthalpy for solidification solver

In the next section, we use the AD-produced sensitivities as a gradient in a data fitting (optimization) process.

5 Optimization Results

In this section we describe the results of using AD-produced gradients in Sandia National Laboratories' DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) [7] toolkit to fit data. We make use of the exact solution temperature profile provided in [6] as if it were experimental data to be fit by our model. We allow two of the input parameters (κ and T_l) to vary to see if the response R (Eq. 7) can be minimized. The results are given in Figure 2.

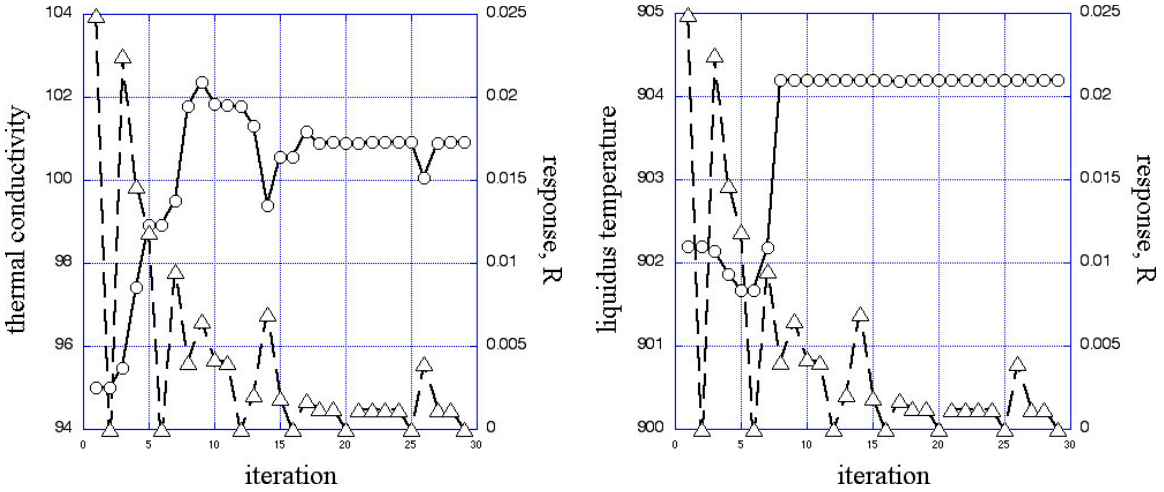

Figure 2: Optimization results for minimizing R by varying κ and T_l

We see that a thermal conductivity increase to 101 W/m-K from its nominal value improves the fit to the "data." The liquidus temperature remains at its nominal value 904.2 K. This is, of course, only a demonstration of the methodology rather than a serious data fitting exercise.

6 Summary and Future Work

We have described a non-linear solver for heat transfer when a binary eutectic system cools and freezes and we have provided accurate AD-produced sensitivities that have, in turn, been used in a data-fitting process. Real problems in three dimensions will certainly have many more parameters describing mold geometry and pouring conditions. If the trend seen here holds true adjoint mode AD is favored since the computational time increases only slightly for additional parameters. The down side of the reverse mode for a non-linear CFD/casting code is that the forward solution must be available for the reverse calculation. Developing an independent-storage/recalculation strategy will be necessary to provide this information when applying this approach to real three-dimensional simulations.

We also will explore the possibility of bypassing the development of a derivative propagating code by means of AD by making direct use of the underlying numerical operators in matrix form. By this means the sensitivities could be propagated in either the forward or reverse (this requires transposing the matrices) modes in a manner similar to the solution. The methods applied in this code should lend themselves to such an approach.

Acknowledgements

Work performed under the auspices of the U. S. Department of Energy at the Los Alamos National Laboratory under Contract W-7405-ENG-36.

References

[1] - , *Telluride Website: Los Alamos National Laboratory Report LA-UR-99-1664,* (http://public.lanl.gov/mww/HomePage.html) (1999).

[2] D. Knoll, D. Kothe, and B. Lally, *A New Nonlinear Solution Method for Phase Change Problems*, Numerical Heat Transfer Part B, 35(4) (June 1999).

[3] - *ADIFOR 3.0 Overview*, Rice University Technical Report CAAM-TR-00-02 (February 2000).

[4] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS publishing Company, Boston, 1996.

[5] J. R. R. A. Martins, I. M. Kroo, and J. J. Alonso, *An Automated Method for Sensitivity Analysis using Complex Variables*, Proceedings of the 38th Aerospace Sciences Meeting, Reno, NV (January 2000), AIAA Paper 2000-0689.

[6] C. R. Swaminathan and V. R. Voller, *Towards a General Numerical Scheme for Solidification Systems*, Int. J. Heat Mass Transfer, 40:2859 (1996).

[7] M. S. Eldred, A. A. Giunta, B.G. van Bloemen Waanders, S. F. Wojtkiewicz, W. E. Hart, and M. P. Alleva, *DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Version 3.0 Users Manual*, Sandia National Laboratories Technical Report SAND2001-3796 (April 2002).